

## II. General Remarks Concerning This Response

Claims 1-30 are currently pending. Claims 1, 2, 14, 15, and 27 have been amended herein; no claims have been added or canceled. Claims 12, 13, 25, 26, and 30 were previously allowed. Claims 9-11 and 22-24 were previously objected to as being allowable if rewritten in independent form. Reconsideration of the claims is requested.

## III. 35 U.S.C. § 102(e)-Anticipation-Heistermann et al.

The Office action has rejected claims 1-8, 14-21, and 27-29 under 35 U.S.C. § 102(e) as anticipated by Heistermann et al., "Version-adaptive serialization and deserialization of program objects in an object-oriented environment", U.S. Patent Number 6,477,701 B1, filed 09/30/1999, issued 11/05/2002. This rejection is traversed.

Applicant asserts that Heistermann et al. cannot be used as an anticipatory reference because Heistermann et al. fails to disclose at least one claim element of the present patent application. Independent claim 1, as amended, reads as follows:

1. A method for object-oriented management of serializable objects, the method comprising:  
identifying an object, wherein the object comprises a set of attributes;  
associating a class version identifier with the object, wherein the class version identifier identifies the object as an instance of a specific version of a class; and  
associating an attribute version identifier with an attribute in the set of attributes such that each attribute in the set of attributes is associated with an attribute version identifier.

The rejection relies on portions of Heistermann et al. at column 6, lines 38-53, and column 7, lines 35-46, for disclosing the third element of claim 1 with respect to attribute version identifiers; these passages read (emphasis added):

FIG. 4 provides additional detail for one particular implementation of this basic serialization process. In this particular implementation, the class of a program object to be serialized defines a method for obtaining the version of the program object. Step 101 instantiates a program object of this class and step 110 invokes the appropriate method to obtain an identification of the program object version. Step 121 identifies the properties for this program object version that should be included in the serialized representation of the program object. This may be implemented in a variety of ways including, for example, a method of the program object to be serialized, a method of a program object in another class that is related in some manner to the class of the program object to be serialized, or a database of properties. In any case, an implementation of a process that determines the properties to serialize may be considered a function of the program object class and version. Step 122 ascertains the value of each property to serialize and **step 130 generates serial information that conveys the program object class and version, and a representation of each respective property that was identified in steps 121 and 122.--**(column 6, lines 32-53).

If this object-descriptor class is derived from an object-descriptor base class, the process continues with step 133 that **writes to the serial information stream an identification of the version pertaining to the base class.** The process continues by serializing the appropriate properties of that base class. This process iterates until step 136 determines that no further base classes remain, at which time step 137 determines whether any further program objects remain to be serialized. If so, the process continues with step 132 for the next respective program object. Otherwise, the process concludes with step 138.--(column 7, lines 35-46).

The above-cited portion of Heistermann et al. shows that the only version identifiers that are written to the serial information stream are the version identifiers for a program object class. In the system of Heistermann et al., the serial information stream contains a representation of a program object class, a version identifier for the program object class, and a representation for each property within the program object class.

However, Heistermann et al. does not disclose a version identifier for each property within a program object class.

In contrast, the present invention recites the following in claim 1, as amended, about attribute version identifiers:

5            associating an attribute version identifier with an attribute in the set of attributes such that each attribute in the set of attributes is associated with an attribute version identifier.

10 Heistermann et al. simply does not disclose an attribute version identifier for each attribute as is claimed by the present application. The attribute version identifiers of the present invention are independent and distinct from a class version identifier, or in the terminology of Heistermann et al., a  
15 version identifier for a program object class. Although the original independent claim 1 of the present application previously recited that an attribute version identifier was associated with an attribute, the amended claim clearly states that each attribute is associated with an attribute version  
20 identifier. Moreover, the attribute version identifiers are clearly independent and distinct from the class version identifier that is recited earlier within the claim.

The Office action rejected independent claims 14 and 27 on similar reasoning, but independent claims 14 and 27 have similar  
25 elements, as amended, with respect to attribute version identifiers as independent claim 1.

With respect to dependent claims 2-8, 13-21, 28, and 29, Heistermann et al. does not disclose, at a minimum, the subject matter in the independent claims from which these dependent  
30 claims depend. Thus, Heistermann et al. also fails to disclose the features of the dependent claims because these dependent claims include the features of independent claims 1, 14, and 27.

Heistermann et al. clearly does not disclose features as required by the claim language. As stated at MPEP § 2131: "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Hence, Heistermann et al. cannot be used as an anticipatory reference, and the rejection of claims 1-8, 14-21, and 27-29 has been overcome, whereby Applicant requests the withdrawal of the rejection.

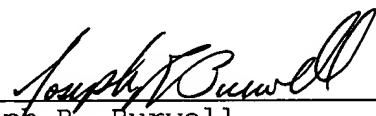
#### IV. Conclusion

It is respectfully urged that the present application is patentable, and Applicant kindly requests a Notice of Allowance.

For any other outstanding matters or issues, the examiner is urged to call or fax the below-listed telephone numbers to expedite the prosecution and examination of this application.

DATE: August 10, 2004

Respectfully submitted,

  
\_\_\_\_\_  
Joseph R. Burwell  
Reg. No. 44,468  
ATTORNEY FOR APPLICANT

Law Office of Joseph R. Burwell  
P.O. Box 28022  
Austin, Texas 78755  
Voice: 866-728-3688 (866-PATENT8)  
Fax: 866-728-3680 (866-PATENT0)  
Email: joe@burwell.biz